DESIGN AND EVALUATION OF DATABASE ACCESS PATHS

by

Christopher D. Keen, B.Sc. (Hons)

Department of Computing Science
University of Adelaide

Thesis submitted for the degree of Doctor of Philosophy

31st January 1978

Awarded 12.3.79.
SUMMARY

This thesis presents an investigation into the analytic modelling of processes which occur in a general purpose database management system. A strictly hierarchical structure is first imposed on the architecture of a database management system. Analytic models are then presented for each level of this hierarchy and algorithms developed to form the interfaces between these levels. Many refinements are made to existing models and algorithms proposed in these areas. The algorithms which are developed form the dynamic components of a database management system and may be directly incorporated into the design of practical systems. The results obtained verify that the chosen levels of modelling are useful both as abstract tools in the understanding of database management theory, and as aids in the design and implementation of large and complex data handling systems.
CONTENTS

CHAPTER 1. INTRODUCTION

1.1. Context of the Investigation 1
1.2. Hierarchical Decomposition of a Database Management System 4
1.3. Objectives of the Investigation 5
1.4. Methods of Investigation 6
1.5. Validation of the Simulation Model 8
1.6. Original Features of the Study 9
1.7. Synopsis of the Thesis 10
1.8. Relational Database Terminology 12

CHAPTER 2. SIMULATION MODEL OF A DATABASE MANAGEMENT SYSTEM

2.1. Introduction 14
   2.1.1. Internal Structure of the Database Management System 16
2.2. Access String Model 20
   2.2.1. Introduction 20
   2.2.2. Access String Declaration 21
   2.2.3. User Queries 25
2.3. Logical Structure of the Simulation Model 26
   2.3.1. Introduction 26
   2.3.2. Hierarchical DBMS Structure 29
   2.3.3. Dependence of the Query Analysis on the Host Operating System 33
   2.3.4. Dependence of the Query Analysis on the Index Structure 37
   2.3.5. Dependence of the Index Structure on the Operating System 40
   2.3.6. Implementation of the DBMS Simulation Model 40

CHAPTER 3. QUERY ANALYSIS

3.1. Introduction 43
3.2. Previous Work 43
3.3. Terminology 45
3.4. Query Term Resolution
3.5. The Query Analysis Algorithm
3.5.1. Construction of a DB Search Strategy
3.5.2. Method of Solution
3.5.3. Choice of Access String Selection Criterion
3.6. Improving the Efficiency of the Query Analysis Algorithm
3.6.1. Relevant Access Strings
3.6.2. Application of Relevant Access Strings
3.6.3. An Upper Bound on the Set of Relevant Strings
3.7. Summary

CHAPTER 4. IMPLEMENTATION OF ACCESS STRINGS

4.1. Mapping of Access Strings into Index Structures
4.1.1. Introduction
4.1.2. Methods of Solving the Index Design Problem
4.1.3. Implementing Partitions
4.2. A General Model of File Retrieval
4.2.1. Introduction
4.2.2. Storage Model
4.2.3. Disc Access Model
4.2.4. Implementation of Files
4.2.5. File Implementation Parameters
4.2.6. List Parameters
4.2.7. Ordered List Retrieval
4.2.7.1. An Example of an Ordered Retrieval
4.2.8. Unordered List Retrieval
4.2.8.1. The LRU-stack Model
4.2.8.2. Page Fault Probability
4.2.8.3. Markov Chain Model of the Buffer’s Contents
4.2.8.4. An Example of an Unordered Retrieval
4.3. Yao’s Method of Optimal Index Design
4.3.1. Introduction
4.3.2. Yao’s Tree Model
4.3.2.1. Implementation of Filial Sets
4.3.3. Database Parameters
4.3.4. Index Tree Structure
  4.3.4.1. Prefix Trees
  4.3.4.2. Sequence Trees
4.3.5. TRIE/TREE Analysis
  4.3.5.1. The TRIE Structure
  4.3.5.2. The TREE Structure
  4.3.5.3. Comparison of TRIE and TREE structures
4.3.6. The Index Design Problem
  4.3.6.1. Complexity of the Index Design Problem
  4.3.6.2. A Mathematical Statement of the TREE Design Problem
4.3.7. Some Shortcomings of Yao's Solution Method

4.4. An Alternative Approach to Index Design
  4.4.1. The Basic Index Design Algorithm
  4.4.2. The Dependence of Variables on the Segment Lengths
  4.4.3. Revised Mathematical Statement of the Index Design Problem
  4.4.4. Cost Analysis of the TREE Structure
  4.4.5. Method of Solution of the Revised Mathematical Programme

4.5. Analysis of Results of the Index Design Algorithm
  4.5.1. The Byte-level Description of the Partitioned Domain
  4.5.2. A Reference Case of the Index Design
  4.5.3. Sensitivity Analysis of the Index Design Algorithm
    4.5.3.1. Variation of the Page Size, PAGLEN
    4.5.3.2. Variation of the Storage Cost Weight, cs
    4.5.3.3. Variation of the Real Time Cost Weight, ct
    4.5.3.4. Variation of the Processing Time Cost Weight, cp
    4.5.3.5. Variation of the Retrieval Frequency, FR
    4.5.3.6. Variation of the Update Frequency, FU
    4.5.3.7. Variation of the Group Factor, GPFAC
    4.5.3.8. Variation of the Internal Buffer Size, BUFLEN
  4.5.4. Summary of Experimental Results
CHAPTER 5. AN ANALYTIC MODEL OF LIST MERGING

5.1. Introduction .................................................. 173
5.2. The List Merge Model ........................................ 173
  5.2.1. Assumptions ............................................ 173
  5.2.2. Union and Intersection ................................. 176
5.3. Merge Strategy ................................................ 176
  5.3.1. Determination of a Merge Tree ......................... 177
  5.3.2. Merge Algorithm ....................................... 179
5.4. Probabilistic Model of a Merge ............................. 181
  5.4.1. List Intersection ....................................... 181
  5.4.2. List Union ............................................... 183
5.5. Estimation of the Number of Disc Accesses ................. 186

CHAPTER 6. CONCLUDING REMARKS

6.1. Aims and Achievements ...................................... 187
  6.1.1. Hierarchical Structure of a Database Management System 187
  6.1.2. Access String Model ................................... 187
  6.1.3. Index Structure ........................................ 188
  6.1.4. Index Design Problem and Solution Method .......... 189
  6.1.5. File and List Model ................................... 190
  6.1.6. Dependence on the Host Operating System .......... 190
  6.1.7. Decomposition of the Database Design Problem .... 191
6.2. Conclusions .................................................. 192
  6.2.1. Suitability of Analytic Modelling .................... 192
  6.2.2. Flexibility through Multilevel Decomposition ...... 193
  6.2.3. Achievement of Generality through Mathematical Analysis 194
6.3. Further Research ............................................ 195
  6.3.1. Detailed Investigation of the Index Design Problem 195
  6.3.2. Extension of the Range of Analytic Models ......... 195
  6.3.3. Specific Design Questions ............................ 196
APPENDIX A.

A.1. Union of Strings
A.2. Intersection of Strings
A.3. String Complement
A.4. Difference of Strings

APPENDIX B.

B.1. Introduction
B.2. Derivation of N(O,NL,NPOP,L)
   B.2.1. Proof of (a)
   B.2.2. Proof of (b)
   B.2.3. Proof of (c)
B.3. Expected Size of List Intersections
   B.3.1. Generalization to Arbitrary List Lengths
B.4. Intersection of n-out-of-NL Lists
   B.4.1. Derivation of S(i,NL,NPOP,L)
   B.4.2. Derivation of Exp(n given EV[LOUT,NL,NPOP,L])
   B.4.3. Alternative Derivation of Exp(n given
          EV[LOUT,NL,NPOP,L])
   B.4.4. Interpretation of Q(n,NL,NPOP,L)
B.5. Approximations to Expected Intersection Sizes
   B.5.1. Limiting Behaviour of the Approximations

REFERENCES.